

A Hybrid Framework for Multi-Vehicle Collision Avoidance

Aparna Dhinakaran*, Mo Chen*, Glen Chou, Jennifer C. Shih, Claire J. Tomlin.

Abstract—With the recent surge of interest in UAVs for civilian services, the importance of developing tractable multi-agent analysis techniques that provide safety and performance guarantees has drastically increased. Hamilton-Jacobi (HJ) reachability has successfully provided these guarantees to small-scale systems and is flexible in terms of system dynamics. However, the exponential complexity scaling of HJ reachability with respect to system dimension prevents its direct application to larger-scale problems where the number of vehicles is greater than two. In this paper, we propose a collision avoidance algorithm using a hybrid framework for $N + 1$ vehicles through higher-level control logic given any N -vehicle collision avoidance algorithm. Our algorithm conservatively approximates a guaranteed-safe region in the joint state space of the $N + 1$ vehicles and produces a safety-preserving controller. In addition, our algorithm does not incur significant additional computation cost. We demonstrate our proposed method in simulation.

I. INTRODUCTION

With the increasingly widespread adoption of unmanned aerial vehicles (UAVs), it is important to understand interactions in multi-vehicle systems; indeed, government agencies such as the Federal Aviation Administration and the National Aeronautics and Space Administration are urgently working on UAV-related regulations [6], [7].

Multi-agent systems are characterized by the asymmetric goals of and cooperation among the involved agents. This setting captures many real-world scenarios, and so these systems has been studied extensively in the past, with a particular focus on collision-avoidance. However, most prior lines of work introduce simplifying assumptions to obtain satisfactory results. For instance, [8], [9] assume that the agents will employ certain simple control strategies which induce velocity obstacles that must be avoided in order to maintain safety. [10], [11] assume that multiple agents maintain formation to travel along pre-specified trajectories. Similarly, other works such as [12]–[16] require strong assumptions on the dynamics of the system. While these approaches capture important behavior of multi-agent systems for many realistic scenarios, they do not offer general safety guarantees important in safety-critical systems.

Attempts at addressing this more general setting have utilized optimal control and differential game approaches. These are ideal tools due to their flexibility with respect to the system dynamics. In particular, the HJ formulation of

differential games has been studied extensively and successfully applied to small-scale problems such as automatic aerial refueling [17], pairwise collision avoidance [18], reach-avoid games [19], [20], and many others. However, despite the many successes of HJ reachability, the computational complexity of HJ-based methods scales exponentially with the number of vehicles in the system, making their direct application to multi-vehicle problems intractable. Thus, utilizing differential games to analyze larger-scale problems usually requires structural assumptions on the roles of different agents or on the traffic rules of the system. For example, [21]–[23] discuss various classes of three-player differential game with different assumptions on the role of each agent in non-cooperative settings. For even larger systems, [24]–[28] provide promising results when varying assumptions in the form of traffic rules and vehicle priorities can be made.

In this paper, we attempt to resolve the above two issues simultaneously; we attempt to solve the problem of collision avoidance in unstructured flight with low computational overhead. Our proposed algorithm achieves collision avoidance for $N + 1$ vehicles via black box use of any algorithm for N -vehicle collision avoidance. In particular, our algorithm employs a hybrid systems approach to provide a joint co-operative control strategy and conservatively approximates a guaranteed-safe region in the joint state space of the $N + 1$ vehicles. To do this, our algorithm treats one of the vehicles as an “outsider” vehicle, and then quantifies the set of safe states for the outsider vehicle given the states of the other N vehicles. To preemptively account for cases in which no safe states can be found, we formulate our solution to allow sufficient time for any vehicle to remove itself from the joint system.

Our paper is organized as follows:

- In Section II, we formulate the multi-vehicle collision avoidance problem, provide a summary of HJ reachability from [18], [29], and state our goals.
- In Section III, we describe our hybrid framework for achieving $N + 1$ -vehicle collision avoidance given some N -vehicle collision avoidance algorithm. We prove that our proposed hybrid framework conservatively approximates the unsafe regions of any vehicle given the other N vehicles, and that the cooperative controllers synthesized by the hybrid framework guarantee safety whenever possible.
- Finally, in Section V, we demonstrate our proposed method in a four-vehicle simulation. (Prior HJ reachability-based approaches for guaranteeing safety were limited to two or three vehicles [18], [29]).

This research is supported by ONR under the Embedded Humans MURI (N00014-16-1-2206).

* Both Authors contributed equally to this work. All authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. {aparnadhinak,mochen72,gchou,csihh,tomlin}@berkeley.edu

II. PROBLEM FORMULATION

Consider $N + 1$ vehicles, denoted $Q_i, i = 1, 2, \dots, N + 1$, whose dynamics are described by the system of ordinary differential equation (ODE)

$$\dot{x}_i = f_i(x_i, u_i), \quad u_i \in \mathcal{U}_i, \quad i = 1, \dots, N + 1 \quad (1)$$

where $x_i \in \mathbb{R}^{n_i}$ is the state of the i^{th} vehicle Q_i , and u_i is the control of Q_i . Each vehicle Q_i will have some objective, such as getting to a set of goal states. Whatever the objective may be, each vehicle Q_i must at all times avoid the *danger zone* \mathcal{Z}_{ij} with respect to each of the other vehicles $Q_j, j \neq i$. These danger zones represent undesirable relative configurations between Q_i and Q_j (such as collisions).

If possible and desired, each vehicle uses a “goal-satisfaction controller” that helps complete its objective. However, sometimes a “safety controller” must be used in order to prevent the vehicle from entering danger zones. Since these danger zones \mathcal{Z}_{ij} are sets of joint configurations, it is convenient to derive the set of relative dynamics between every vehicle pair from the dynamics of each vehicle specified in (1). Let the relative dynamics between Q_i and Q_j be specified by the following ODE.

$$\begin{aligned} \dot{x}_{ij} &= g_{ij}(x_{ij}, u_i, u_j) \\ u_i &\in \mathcal{U}_i, u_j \in \mathcal{U}_j \quad i, j = 1, \dots, N + 1, i \neq j \end{aligned} \quad (2)$$

We assume the functions f_i and g_{ij} are uniformly continuous, bounded, and Lipschitz continuous in arguments x_i and x_{ij} respectively for fixed u_i and (u_i, u_j) respectively. In addition, the control functions $u_i(\cdot) \in \mathbb{U}_i$ are drawn from the set of measurable functions¹.

A. Background

1) *Hamilton-Jacobi Reachability*: The main goal of reachability analysis is to characterize the set of states either from which the system can be driven into some target set, or that the system can reach from some target set. Specifically, in the backward reachability problem, we would like to determine the backward reachable set (BRS) $\mathcal{V}(T)$ of time horizon T representing states from which the system can be driven into some potentially *time-varying* target set $\mathcal{L}(t), t \in [-T, 0]$. In the forward reachability problem, we would like to determine the forward reachable set (FRS) $\mathcal{W}(T)$ of time horizon T representing states that a system can reach starting from the target set \mathcal{L} . In general, $\mathcal{V}(t)$ can be obtained from the viscosity solution [30] to the HJ PDE (3). Although there are many equivalent methods of obtaining the BRS [18], [31], [32], we will use the formulation in [32] which solves the time-varying reachability problem without augmenting the state space with time.

$$\begin{aligned} \min\{D_t V(t, x) + H(x, \nabla V(t, x)), l(t, x) - V(t, x)\} &= 0 \\ V(T, x) &= l(T, x) \\ t &\in [0, T] \end{aligned} \quad (3)$$

¹A function $f : X \rightarrow Y$ between two measurable spaces (X, Σ_X) and (Y, Σ_Y) is said to be measurable if the preimage of a measurable set in Y is a measurable set in X , that is: $\forall V \in \Sigma_Y, f^{-1}(V) \in \Sigma_X$, with Σ_X, Σ_Y σ -algebras on X, Y .

where the target set is represented by the sub-zero level set of the implicit surface function $l(x)$: $\mathcal{L} = \{x : l(x) \leq 0\}$. The value function $V(T, x)$ is the implicit surface function representing the BRS: $\mathcal{V}(T) = \{x : V(T, x) \leq 0\}$.

In a similar fashion, the FRS $\mathcal{W}(T)$ can be computed with the following HJI PDE:

$$\begin{aligned} D_t W(t, x) + H(x, \nabla W(t, x)) &= 0 \\ W(0, x) &= l(x) \\ t &\in [0, T] \end{aligned} \quad (4)$$

where the target set $\mathcal{L} = \{x : l(x) \leq 0\}$ specifies the initial set of states, the FRS $\mathcal{W}(T) = \{x : W(T, x) \leq 0\}$ is given by the value function, and the Hamiltonian H depends on the role of the control inputs.

2) *Potential Conflict*: Intuitively, two vehicles are in potential conflict when they are likely to enter each other’s danger zones. This indicates when a vehicle will employ the safety controller as opposed to using the goal satisfaction controller. This notion is formally stated in the following definition of BRS $\mathcal{V}_{ij}^{PC}(T)$ (PC for potential conflict), Hamiltonian, and optimal avoidance control for vehicle Q_i :

Definition 1: The **Potential Conflict Set**, P_{ij} , is the set of relative states x_{ij} such that if $x_i \notin P_{ij}$, then Q_i is guaranteed to be able to avoid \mathcal{Z}_{ij} with Q_j , no matter what non-anticipative control strategy Q_j uses [18].

$$\begin{aligned} \mathcal{V}_{ij}^{PC}(T) &= \{y : \forall u_i, \exists u_j, \exists t \in [0, T], \\ &\quad x_{ij}(\cdot) \text{ satisfies (2), } x_{ij}(t) = y \in \mathcal{Z}_{ij}\} \end{aligned} \quad (5)$$

$$H_{ij}^{PC}(x_{ij}, \lambda) = \max_{u_i} \min_{u_j} \lambda \cdot g_{ij}(x_{ij}, u_i, u_j) \quad (6)$$

$$u_i^{PC}(x_{ij}) = \arg \max_{u_i} \min_{u_j} \nabla V_{ij}^{PC}(x_{ij}) \cdot g_{ij}(x_{ij}, u_i, u_j) \quad (7)$$

where $V_{ij}^{PC}(x_{ij}) := \lim_{T \rightarrow \infty} V_{ij}^{PC}(x_{ij}, T)$.

$$P_{ij} = \{x_i : V_{ij}^{PC}(x_{ij}) \leq K\} \quad (8)$$

When N vehicles are in potential conflict, then an N -vehicle collision avoidance algorithm is needed to ensure their safety.

Remark 1: Unlike \mathcal{V}_{ij}^{PC} , P_{ij} is defined in the space of x_i instead of x_{ij} . This is done for convenience in latter parts of the paper.

B. Goals of this Paper

Consider a scenario in which at time $t = 0$, a set of $N + 1$ vehicles are in potential conflict with each other; in other words, the “total conflict size²” is $N + 1$. Since the N -vehicle collision avoidance algorithm can guarantee safety for N of the $N + 1$ vehicles, we can choose N vehicles to be handled by such a collision avoidance algorithm, and derive conditions on the remaining vehicle under which the entire system of $N + 1$ vehicles can remain safe. To account for the situation in which safety of the entire system is impossible, we will assume that within an “exit time” T_e , one of the vehicles will be able to remove itself from the system.

²The notion of conflict size via determining potential conflicts among multiple vehicles will be formally defined later in Definition 3.

Given the vehicle dynamics in (1), some joint objective, a previously chosen N -vehicle collision avoidance algorithm, the derived relative dynamics in (2), and the danger zones \mathcal{Z}_{ij} , we propose a cooperative safety control strategy that performs the following:

- 1) Detect potential conflicts based on the joint configuration of $N + 1$ vehicles, and in particular determine the size of conflict;
- 2) Conservatively determine without significant computation cost when all $N + 1$ vehicles are guaranteed to be safe, and provide safety controllers for all $N + 1$ vehicles in this case;
- 3) Determine when safety cannot be guaranteed, in which case a vehicle needs to be removed from the system.

We will show that our analysis provides with low overhead a conservative approximation of various slices of a high-dimensional BRS representing the unsafe region in the joint state space of all $N + 1$ vehicles. Such a high-dimensional BRS is intractable to compute directly.

III. METHODOLOGY

In this section, we illustrate our proposed hybrid framework that ensures safety of $N + 1$ vehicles. We will first define terminologies essential to our hybrid framework.

Definition 2: T_e -buffer set.

$$\mathcal{V}_{ij}^E = \{y : \exists u_i, \exists u_j, \exists t \in [0, T_e], x_{ij}(\cdot) \text{ satisfies (2), } x_{ij}(t) = y \in \mathcal{Z}_{ij}\} \quad (9)$$

The purpose of the T_e -buffer set is to account for the situation in which our method indicates that $N + 1$ -vehicle potential conflicts cannot be guaranteed to be resolved safely. The T_e -buffer set allows a vehicle a duration of T_e to remove itself safely from the system; this can be done by maneuvers such as exiting the altitude range. To provide the last-resort option of removing a vehicle from the system, we specify \mathcal{Z}_{ij} to be the T_e -buffer set.

Therefore, if x_i is on the boundary of P_{ij} , then Q_i is said to be “in potential conflict” with Q_j , and the control $u_i^{PC}(x_{ij})$ must be used by Q_i in order to ensure that the relative state of Q_i and Q_j does not enter the T_e -buffer set.

Definition 3: Conflict size. Consider a graph in which each vehicle Q_i is a node. We connect two nodes Q_i, Q_j with an edge if and only if Q_i is in potential conflict with Q_j , vice versa, or both. The number of nodes with at least one edge in the resulting graph is defined as the conflict size.

A. Safety of the Outsider

In this paper, we assume that at $t = 0$, the conflict size first becomes at least N . When this happens, N of the vehicles will perform avoidance maneuvers according to the N -vehicle collision avoidance algorithm. Let T_r be the time it takes for the N -vehicle collision avoidance algorithm to reduce the conflict size *among these N vehicles* by at least 1. We call T_r the “time of resolution.” Note that T_r needs not be known *a priori*, and may be determined at $t = 0$.

1) Unsafe Region: We let \mathcal{N} denote the set of indices of the N vehicles performing avoidance maneuvers according to the N -vehicle collision avoidance algorithm and let I denote the index of the “outsider vehicle.” The N vehicles whose indices are in the set \mathcal{N} are guaranteed to be safe with respect to each other. For the safety of all $N + 1$ vehicles, it remains to ensure that Q_I , the “outsider vehicle” not in \mathcal{N} , is also able to avoid the T_e -buffer set of all other vehicles. The following proposition provides an intuitive condition under which the safety of all $N + 1$ vehicles can be guaranteed.

Proposition 1: If $Q_i, i \in \mathcal{N}$ are in potential conflict and engaged in an N -way collision avoidance, then all $N + 1$ vehicles are guaranteed to be able to avoid the T_e -buffer set $\mathcal{V}_{ij}^E(T_e)$ if Q_I is in potential conflict with a maximum one of the other N vehicles during $t \in [0, T_r]$.

Proof: Since the N -way collision avoidance algorithm guarantees that no vehicle $Q_i, i \in \mathcal{N}$ enter each other’s T_e -buffer set $\mathcal{V}_{ij}^E(T_e)$, it remains to ensure that the outsider vehicle Q_I does not enter any of the other vehicle’s T_e -buffer set. If Q_I is not in potential conflict with any vehicle, then it is safe. And if Q_I is in potential conflict with exactly one other vehicle, say Q_1 without loss of generality, then applying the optimal control $u_I^{APC}(x_{I1})$ would guarantee that $\mathcal{V}_{I1}^E(T_e)$ is not entered. ■

Suppose that to resolve the N -way conflict, a N -way collision avoidance algorithm stipulates some trajectories $x_i(t), i \in \mathcal{N}, t \in [0, T_r]$. To conservatively approximate the safe regions of Q_I , we must ensure that the outsider vehicle cannot be in potential conflict with more than one vehicles until the N -vehicle conflict is resolved. We now define the region in which Q_I is in potential conflict with multiple vehicles.

Definition 4: Outsider unsafe region.

$$\mathcal{S}_I(t) = \bigcup_{i,j \in \mathcal{N}, i \neq j} (\bar{P}_{Ii}(t) \cap \bar{P}_{Ij}(t)) \quad (10)$$

Note that $\mathcal{S}_I(t)$ is a set of states in the space of x_i, \mathbb{R}^{n_i} , and that $t \in [0, T_r]$.

If we can guarantee that Q_I never enters $\mathcal{S}_I(t)$ for $t \in [0, T_r]$, then the following control strategy would keep Q_I from entering the T_e -buffer sets of all other vehicles:

$$u_I(x_I, t) \in \begin{cases} \{u_I^{APC}(x_{Ij}(t))\} & \text{if } x_I \in \bar{P}_{Ij}(t) \\ \mathcal{U} & \text{otherwise} \end{cases} \quad (11)$$

For convenience, we will denote the set of control functions satisfying the control given in (11) for $t \in [0, T_r]$ to be \mathbb{U}_I^{SC} (SC for single conflict). Thus if $u_I(\cdot) \in \mathbb{U}_I^{\text{SC}}$, and $x_I(t) \notin \mathcal{S}_I(t) \forall t \in [0, T_r]$, then Q_I would be guaranteed to not enter into the T_e -buffer zones of any other vehicle. We now proceed to describe how to ensure the above conditions are met.

2) Optimally Avoiding the Unsafe Region: For Q_I to avoid $\mathcal{S}_I(t)$, we must ensure that Q_I avoids the set of states that inevitably leads to $\mathcal{S}_I(t), t \in [0, T_r]$. This set is captured by the following BRS:

Definition 5: Minimal BRS from $\mathcal{S}_I(t)$.

$$\mathcal{V}_I^-(t) = \{y : \forall u_I(\cdot) \in \mathbb{U}_I^{\text{SC}}, \exists s \in [t, T_r], x_I(\cdot) \text{ satisfies (1), } x_I(s) = y \in \mathcal{S}_I(s)\} \quad (12)$$

This BRS can be computed using the $\mathcal{S}_I(t)$ as a time-varying target by solving (3) with the Hamiltonian

$$H_I^-(t, x_I, \lambda) = \begin{cases} \lambda \cdot f(x_I, u_I^{\text{APC}}(x_{Ij}(t))) & \text{if } x_I \in \bar{P}_{Ij}(t) \\ \max_{u_I \in \mathcal{U}_I} \lambda \cdot f(x_I, u_I) & \text{otherwise} \end{cases} \quad (13)$$

$$u_I^-(x_I, t) = \arg \max_{u_I \in \mathcal{U}_I} \nabla V_I^-(x_I, t) \cdot f(x_I, u_I) \quad (14)$$

Proposition 2: The following controller guarantees that if $x_I(0) \notin \mathcal{V}_I^-(0)$, then $x_I(t) \notin \mathcal{V}_I^-(t) \forall t \in [0, T_r]$.

$$u_I(x_I, t) \in \begin{cases} \{u_I^-(x_I, t) \text{ in (14)}\} & \text{if } x_I \in \partial V_I^-(x_I, t) \\ \mathcal{U}_I & \text{otherwise} \end{cases} \quad (15)$$

Proposition 2 not only provides a safety-preserving controller for Q_I . Along with the definition of $V_I^-(x_I, t)$, but also represents a conservative guaranteed-safe region for Q_I , given the avoidance maneuvers of $Q_i, i \in \mathcal{N}$. Depending on the system dynamics and computation power available, if $V_I^-(x_I, t)$ is computable in real-time, then $V_I^-(x_I, t)$ can be used to indicate a “fenced off” region that Q_I must avoid in order to ensure the safety of the $(N + 1)$ -vehicle system.

3) *Fast Safety Verification:* When online computation resources are scarce, then we can consider the following BRS instead.

Definition 6: Maximal BRS from \mathcal{S}_I .

$$\mathcal{V}_I^+(t) = \{y : \exists u_I(\cdot), \exists s \in [t, T_r], x_I(\cdot) \text{ satisfies (1), } x_I(s) = y \in \mathcal{S}_I(s)\} \quad (16)$$

Note that since the only difference between $\mathcal{V}_I^-(t)$ and $\mathcal{V}_I^+(t)$ is the quantifier in front of u_I , we have $\mathcal{V}_I^-(t) \subseteq \mathcal{V}_I^+(t)$, and therefore staying out of $\mathcal{V}_I^+(t)$ is sufficient for guaranteeing safety.

Although computing $\mathcal{V}_I^+(t)$ in general is just as difficult as computing $\mathcal{V}_I^-(t)$, checking whether a state x_I is in $\mathcal{V}_I^+(t)$ can be done much faster, without actually computing $\mathcal{V}_I^+(t)$. This is achieved through the following FRS in $t \in [0, T_r]$:

Definition 7: Forward reachable set from x_I .

$$\mathcal{W}_I(t) = \{y : \exists u_I(\cdot), x_I(\cdot) \text{ satisfies (1), } x_I(0) = x_{I0}, x_i(t) = y\} \quad (17)$$

This FRS can be computed offline assuming that x_{I0} is at the origin. Vehicle dynamics usually do not depend on the position of the vehicle. In this case, the FRS can be simply transformed to the coordinate system in which the vehicle is at the origin of the state space. Hence, no significant online computation is required. We now state the equivalence of checking the membership of x_{I0} in the BRS $\mathcal{V}_I^+(0)$ and checking membership of x_{I0} in the FRS $\mathcal{W}_I(t), t \in [0, T_r]$.

Proposition 3:

$$x_I(0) \in \mathcal{V}_I^+(0) \Leftrightarrow \exists t \in [0, T_r], \mathcal{W}_I(t) \cap \mathcal{S}_I(t) \neq \emptyset \quad (18)$$

Proof:

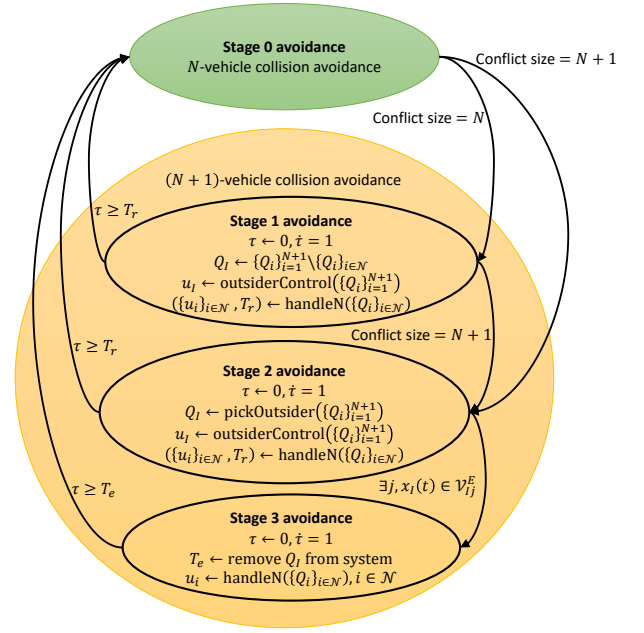


Fig. 1: Summary of Hybrid Framework. The stages of avoidance are explained in Section III-B.

Suppose $x_I(0) \in \mathcal{V}_I^+(0)$, then equivalently

$$\exists u_I(\cdot), \exists t \in [0, T_r], x_I(\cdot) \text{ satisfies (1), } x_I(t) = y \in \mathcal{S}_I(t) \quad (19)$$

Since all quantifiers are existential, we equivalently have

$$\exists t \in [0, T_r], \exists u_I(\cdot), x_I(\cdot) \text{ satisfies (1), } x_I(t) = y \in \mathcal{S}_I(t) \quad (20)$$

By Definition 7, we in turn equivalently have

$$\exists t \in [0, T_r], y \in \mathcal{W}_I(t) \quad (21)$$

But we also have $y \in \mathcal{S}_I(t)$. Therefore, equivalently, $\mathcal{W}_I(t) \cap \mathcal{S}_I(t) \neq \emptyset$. ■

B. Hybrid Framework

Based on the reachability analysis of outsider safety, we propose a hybrid framework that determines control for the $N + 1$ vehicles depending on the conflict size. Our proposed framework provides several stages of safety checking. The framework is depicted in Fig. 1.

Note that all the reachable sets are either defined in the relative coordinates between two vehicles, or in the absolute coordinate with respect to one vehicle, and thus the system dimensionality remains low, allowing for tractable HJ reachability computations. In addition, with the exception of $\mathcal{V}_I^-(t)$, the minimal BRS from $\mathcal{S}_I(t)$, all computations can be done offline and the resulting reachable sets can be stored as lookup tables. This way, we do not incur significant online computational burden.

1) *Stage 0 Avoidance:* When the conflict size is less than N , the vehicles in potential conflict can be managed by the N -vehicle collision avoidance algorithm, and all vehicles are guaranteed safe.

2) *Stage 1 Avoidance*: When the conflict size is N , we have a clear outsider vehicle Q_I that is not in conflict with other vehicles $Q_i, i \in \mathcal{N}$. Our hybrid controller preemptively attempts to keep Q_I from being in potential conflict with more than one of the other N vehicles during $t \in [0, T_r]$. Algorithm 1 provides a safety controller strategy for Q_I .

3) *Stage 2 Avoidance*: During Stage 2 avoidance, the size of conflict is $N+1$. To attempt to ensure safety, we determine whether there exists a vehicle that can be designated as Q_I , such that it is guaranteed to only need to avoid one of the other N vehicles for a duration of T_r . Algorithm 2 outlines the process of designating a vehicle as the outsider vehicle in detail. After the outsider is designated, the safety control for Q_I is given according to Algorithm 1.

If Q_I does enter $S_I(t)$ during $t \in [0, T_r]$, no guarantees on avoiding the T_e -buffer sets can be made; however, Q_I may be able to avoid multiple vehicles with the same control. If at time t before T_r , $x_I(t) \in \mathcal{V}_{Ij}^E$, then we enter Stage 3 avoidance. Else, at T_r , we enter Stage 0.

4) *Stage 3 Avoidance*: Q_I is removed from the system, and the total number of vehicles is reduced to N , allowing the N -vehicle collision avoidance algorithm to maintain safety. The removal of the vehicle can be done by, for example, exiting the altitude range.

C. Outsider Vehicle Control Strategy

Algorithm 1: outsiderControl(Q_1, \dots, Q_{N+1})

```

totalVehicleSet =  $\{Q_1, \dots, Q_{N+1}\}$ 
nVehicleSet = totalVehicleSet -  $Q_I$ 
 $[T_r, NTraj]$  = handleN(nVehicleSet)
 $S_I$  = computeUnsafeRegions( $T_r$ , NTraj)
if  $S_I \cap \mathcal{W}_I(t) == \emptyset \forall t \in [0, T_r]$  then
    Return  $u_I(x_I, t)$  from (11)
else
     $\mathcal{V}_I^-(T_r)$  = getBRS( $S_I$ )
    Return  $u_I(x_I, t)$  from (15)
end if

```

The $N+1$ collision avoidance algorithm is guaranteed to be safe if Q_I is guaranteed to be able to avoid $S_I(t)$. To determine $S_I(t)$, the trajectories $x_i(t), i \in \mathcal{N}, t \in [0, T_r]$ are needed. We assume that these trajectories along with T_r are given by the N -vehicle collision avoidance algorithm, which we call “handleN()” in Algorithm 1.

Algorithm 1 first tries the fast safety verification method outlined in Section III-A.3 to check if Q_I is guaranteed to avoid $S_I(t)$. If Q_I is outside the maximal BRS, $\mathcal{V}_I^+(T_r) \forall t \in [0, T_r]$, then Q_I is guaranteed safe. As proved earlier, this can be efficiently checked using the FRS: If $\mathcal{W}_I(t)$ does not intersect with $S_I(t) \forall t \in [0, T_r]$, then Q_I is guaranteed to be safe. By the definition of $\mathcal{W}_I(t)$, regardless of Q_I ’s actions, it will not need to avoid more than one vehicle in the set $\{Q_i\}, i \in \mathcal{N}$. In this case, (11) specifies the control strategy for Q_I .

If there is an intersection between $\mathcal{W}_I(t)$ and $S_I(t)$ for some $t \in [0, T_r]$, we can still check if Q_I is outside the minimal BRS $\mathcal{V}_I^-(0)$ from $S_I(t)$. $\mathcal{V}_I^-(0)$ specifies the set of states outside of which Q_I is guaranteed to have a control strategy to not enter $S_I(t)$. If $x_I(0) \notin \mathcal{V}_I^-(0)$, then (15) specifies the safety-preserving control strategy for Q_I .

If $x_I(t) \in \mathcal{V}_I^-(0)$, then Q_I is not guaranteed to avoid $S_I(t)$. In this case, it may be possible for Q_I to enter a T_e -buffer zone, in which case Q_I will be safely removed from the system.

D. Designating a Suitable Outsider

If the system enters Stage 2 avoidance, a vehicle needs to be designated as the outsider vehicle Q_I . This is achieved by iterating through each vehicle Q_i where $i \in \{1 \dots N+1\}$ and determining whether safety guarantees can be made if Q_i is the outsider vehicle. Algorithm 2 describes this process in detail.

Algorithm 2: pickOutsider(Q_1, \dots, Q_{N+1})

```

totalVehicleSet =  $\{Q_1, \dots, Q_{N+1}\}$ 
leastConflictVehicles = getLeastConflictVehicles()
for  $Q_I \in$  leastConflictVehicles do
    nVehicleSet = totalVehicleSet -  $Q_I$ 
     $[T_r, NTraj]$  = handleN(nVehicleSet)
     $S_I$  = computeUnsafeRegions( $T_r$ , NTraj)
    if  $S_I \cap \mathcal{W}_I(t) == \emptyset \forall t \in [0, T_r]$  then
        Return  $Q_I$ 
    else
         $\mathcal{V}_I^-(T_r)$  = getBRS( $S_I$ )
        if  $Q_I \notin \mathcal{V}_I^-(T_r)$  then
            Return  $Q_I$ 
        end if
    end if
end for
Return any  $Q_i, i \in \{1 \dots N+1\}$ 

```

While picking a suitable outsider vehicle, we want to have the N vehicle collision avoidance algorithm resolve as many conflicts as possible. Therefore, we select the vehicle with the least conflicts with the other vehicles. After the exclusion of this vehicle, the N -vehicle collision avoidance algorithm can resolve the conflicts of the rest of the N vehicles.

Since we have chosen the outsider vehicle to be the one with the least number of conflicts, the remaining N vehicles will still have a conflict size of at least one, and the N -vehicle collision avoidance algorithm would then reduce this conflict size by at least 1 within a duration of T_r .

E. Communication Scheme

The $N+1$ collision avoidance algorithm does not require a centralized communication architecture. For each possible outsider Q_I , as long as the trajectory of each of the Q_i where $i \in \mathcal{N}$ is broadcast, Q_I can determine if safety is guaranteed. Once an outsider Q_I is determined, its control strategy can be determined with the broadcast trajectories. Note that a

protocol needs to be established to avoid assigning multiple outsider vehicles; for example, the first vehicle to declare itself as an outsider could be chosen, as long as it is a valid choice according to Algorithm 2.

IV. SAFETY GUARANTEE

In the previous section, we proposed a hybrid framework that allows us to detect whether the system of $N + 1$ vehicles is guaranteed to be safe. We now formally state this guarantee and prove the result.

Theorem 1: Suppose there are N vehicles that enter into potential conflict at time $t = 0$. Denote the outsider vehicle as Q_I . If Q_I does not enter $S_I(t)$ for $t \in [0, T_r]$, then we are guaranteed safety for $N + 1$ vehicles for a duration of T_r .

Proof: To prove our theorem, it suffices to prove the following points:

The set P_{Ij} refers to the locations of potential conflict between Q_I and Q_j . If we assume that Q_j is part of the N vehicle collision avoidance algorithm, then our safety controller is assumed to only guarantee safety when Q_I can avoid one vehicle at a time. Therefore, if Q_I is in the potential conflict set P_{Ij} , Q_I must avoid Q_j since Q_j is occupied with the N -vehicle collision avoidance algorithm.

If Q_I is in the P_{Ij} and P_{Ia} , it is in potential conflict with Q_j and Q_a . If Q_I is able to avoid both for T_r , then perhaps Q_I can be guaranteed safety from both Q_j and Q_a . However, it is not always guaranteed that Q_I can avoid both Q_j and Q_a with the same control. The set $S_I(t)$ for $t \in [0, T_r]$ is the set of locations where Q_I will be in potential conflict with more than 1 vehicle. Therefore, we can guarantee safety if Q_I does not enter $S_I(t)$ for $t \in [0, T_r]$ because Q_I is guaranteed to be able to avoid at most one vehicle at a time. Note that we have effectively computed a conservative approximation of a very high-dimensional reachable set by determining unsafe regions. ■

Theorem 2: If Q_I is in $S_I(t)$ for some $t \in [0, T_r]$, Q_I may still be able to avoid the other vehicles as long as it does not enter the \mathcal{V}_{ij}^E of any Q_j . If Q_I enters \mathcal{V}_{ij}^E of any Q_j , Q_I is forced to exit altitude range and is guaranteed to not collide with another vehicle.

Proof: For a particular configuration, if there does not exist a Q_I that can avoid $\mathcal{V}_I^-(t)$ for all $t \in [0, T_r]$, then we are not guaranteed to have a control strategy where $N + 1$ vehicles are guaranteed safe without forcing a vehicle to exit.

However, even if we are unable to find an ideal vehicle Q_I that can avoid $\mathcal{V}_I^-(t)$ for all $t \in [0, T_r]$, we can still employ a vehicle to be our Q_I . At some point t , this Q_I will be in potential conflict with two vehicles, let's call them Q_j and Q_k . It is possible that the avoidance control could avoid Q_j and Q_k at the same time. As long as Q_I does not enter \mathcal{V}_{Ij}^E or \mathcal{V}_{Ik}^E , Q_I is not required to be removed from the system.

If Q_I enters \mathcal{V}_{Ij}^E or \mathcal{V}_{Ik}^E , then it must be removed from the system. By definition of \mathcal{V}_{ij}^E , as long as Q_I is able to be removed from the system within a duration of T_e , safety can be guaranteed. ■

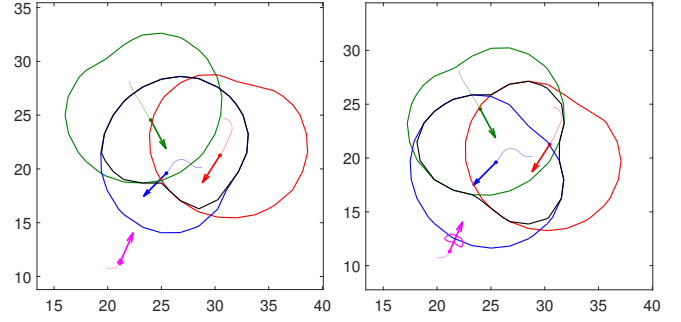


Fig. 2: Stage 1: A scenario where the FRS of the outsider vehicle (pink) does not intersect with the outsider unsafe region (black) for the entire time horizon that the rest of the three vehicles take to resolve their conflicts.

V. NUMERICAL SIMULATIONS

In this section, we illustrate our method through simulations of a four-vehicle system, building our $N + 1$ vehicle framework on top of the three-vehicle collision avoidance MIP scheme. We assume that the dynamics of each vehicle Q_i are identical and described by the Dubins car model:

$$\begin{aligned}\dot{p}_{x,i} &= v \cos(\theta_i) \\ \dot{p}_{y,i} &= v \sin(\theta_i) \\ \dot{\theta}_i &= \omega_i, |\omega_i| \leq \bar{\omega}\end{aligned}\quad (22)$$

The states $p_{x,i}, p_{y,i}, \theta_i$ represent the x , y , and θ coordinates respectively of vehicle Q_i . The danger zone for HJ computation between Q_i and Q_j is defined as

$$\mathcal{Z}_{ij} = \{x_{ij} : (p_{x,i} - p_{x,j})^2 + (p_{y,i} - p_{y,j})^2 \leq R_c^2\} \quad (23)$$

x_{ij} refers to the relative coordinates between Q_i and Q_j : $x_{ij} = [p_{x,ij}, p_{y,ij}, p_{\theta,ij}]^\top = [p_{x,i} - p_{x,j}, p_{y,i} - p_{y,j}, \theta_i - \theta_j]^\top$.

In our experiment, each vehicle travels at a fixed speed of $v = 1$ and its maximum turn rate is $\bar{\omega} = 1$. The time for a vehicle to exit T_e is set to 2; the collision radius R_c for each danger zone \mathcal{Z}_{ij} is 3; and the safety threshold K is 2.0.

To obtain the optimal pairwise safety controller, we compute the BRS $\mathcal{V}_{ij}^{PC}(T)$ with relative dynamics

$$\begin{aligned}\dot{p}_{x,ij} &= -v + v \cos(\theta_{ij}) + \omega_i p_{y,ij} \\ \dot{p}_{y,ij} &= v \sin(\theta_{ij}) - \omega_i p_{x,ij} \\ \dot{\theta}_{ij} &= \omega_i - \omega_j, |\omega_i|, |\omega_j| \leq \bar{\omega}\end{aligned}\quad (24)$$

We present the simulation for Stage 1 and 2 in detail when number of vehicles $N = 4$. Each vehicle aims to reach the circular target of matching color while avoiding the danger zones of the other vehicles.

In Fig. 2, we illustrate the scenario where the FRS of the outsider vehicle (pink) does not overlap with the outsider unsafe region 10 (black) in stage 1. The potential conflict zones are plotted for the three vehicles in their colors. In particular, we show two subplots illustrating the FRS and the outsider unsafe region at a time horizon of $T = 2$ and $T = 27$ respectively. In this case, the outsider vehicle is free to use the goal-satisfaction controller to go to its target

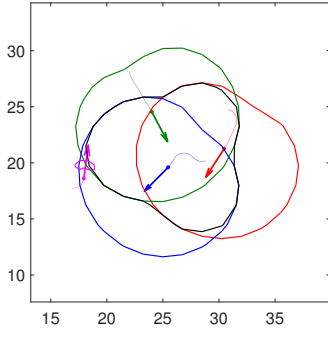


Fig. 3: A stage 1 scenario where the FRS of the outside vehicle (pink) intersects with the outsider unsafe region (black).

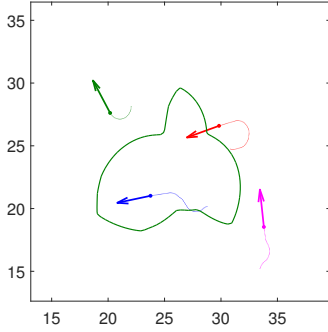


Fig. 4: A stage 2 avoidance scenario in which the green vehicle is chosen as the outsider; the green boundary shows that it is outside of $\mathcal{V}_I^-(t)$ and thus can avoid getting into multiple conflicts.

while the rest of the three vehicles resolve their potential conflicts. In Fig. 3, we show a scenario where the FRS of the outsider vehicle intersects with the outsider unsafe region. In this case, we compute the BRS of this region and the outsider vehicle preemptively executes the optimal controller to avoid this region while the rest of the three vehicles resolve conflicts. Note that in this graph, we're taking the 2D projection of the reachable sets so it looks like the outsider vehicle is inside the potential conflict zone of another vehicle though the outsider vehicle is outside of the potential conflict zone in the state space.

Fig. 4 shows a stage 2 scenario in which the green vehicle is chosen as the outsider. The green boundary represents the a 2D slice of the BRS $\mathcal{V}_I^-(t)$ taken at the green vehicle's current heading, and contains the set of positions from which the green vehicle would inevitably enter the set $\mathcal{S}_I(t)$. Since the green vehicle is outside of $\mathcal{V}_I^-(t)$, it is guaranteed to be able to avoid $\mathcal{S}_I(t)$, thereby avoid getting into multiple conflicts with the other three vehicles. Thus, if the green vehicle uses the controller $u_I^-(x_I, t)$ in (5), the entire conflict can be resolved, as shown in Fig. 5.

The BRS of the N -vehicle potential conflict intersections is computed online. The other BRS and FRS can be computed offline and reused. Each BRS and FRS takes approximately 1 minute to compute. All computations were

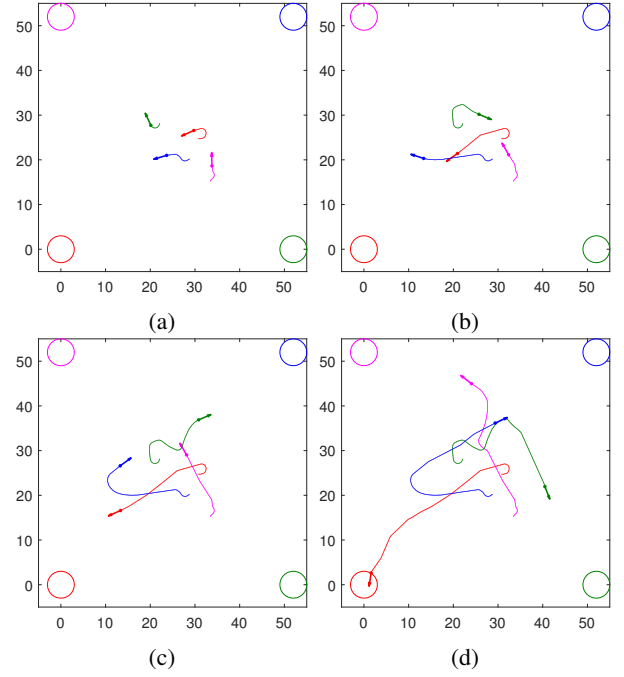


Fig. 5: Snapshots in time of a four-vehicle simulation in which stage 2 avoidance occurs (the same simulation as that shown in Fig. 4). The green vehicle is chosen as the outsider and it does not interfere with the avoidance maneuvers of the other three vehicles. The colored circles represent the vehicles' destinations.

done on a MacBookPro 11.2 laptop with an Intel Core i7-4750 processor.

VI. CONCLUSIONS

We considered the problem of $N + 1$ -vehicle collision avoidance using a hybrid framework. By exploiting properties of pairwise optimal collision avoidance and an existing N -vehicle collision algorithm, our proposed method conservatively approximates the unsafe region of any vehicle given the states of the other N vehicles, and synthesizes a cooperative control strategy that guarantees collision avoidance of all $N + 1$ vehicles whenever possible as determined by our hybrid framework. In the situation in which collision avoidance cannot be guaranteed for all $N + 1$ vehicles, our proposed method still allows enough time for any vehicle to remove itself from the system.

Immediate future work includes better preemptive avoidance to allow prediction of the unsafe regions so that collision avoidance can be guaranteed in all scenarios. Other future work includes investigating safety guarantees for a larger number of vehicles, and incorporating more collision avoidance algorithms into our proposed framework.

REFERENCES

- [1] Amazon.com, Inc. (2016) Amazon prime air. <http://www.amazon.com/b?node=8037720011>. Accessed: Oct. 11, 2016.
- [2] AUVSI News. (2016) Uas aid in south carolina tornado investigation. <http://www.auvsi.org/blogs/auvsi-news/2016/01/29/tornado>. Accessed: Oct. 9, 2016.

- [3] BBC Technology. (2016) Google plans drone delivery service for 2017. <http://www.bbc.com/news/technology-34704868>. Accessed: Oct. 10, 2016.
- [4] W. M. Debusk, "Unmanned aerial vehicle systems for disaster relief: Tornado alley," in *Infotech@Aerospace Conferences*, 2010.
- [5] B. P. Tice, "Unmanned aerial vehicles – the force multiplier of the 1990s," *Airpower Journal*, 1991.
- [6] Jointed Planning and Development Office (JPDO), "Unmanned aircraft systems (UAS) comprehensive plan – a report on the nation's UAS path forward," Federal Aviation Administration, Tech. Rep., 2013.
- [7] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. R. III, "Uas traffic management (utm) concept of operations to safely enable low altitude flight operations," in *AIAA Aviation Technology, Integration, and Operations Conference*, 2016.
- [8] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, pp. 760–772, 1998.
- [9] J. van den Berg, M. C. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *IEEE International Conference on Robotics and Automation*, May 2008, pp. 1928–1935.
- [10] Y.-L. Chuang, Y. Huang, M. R. D'Orsogna, and A. L. Bertozzi, "Multi-vehicle flocking: Scalability of cooperative control algorithms using pairwise potentials," in *IEEE International Conference on Robotics and Automation*, April 2007, pp. 2292–2299.
- [11] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *IFAC World Congress*, 2002.
- [12] A. Ahmadzadeh, N. Motee, A. Jadbabaie, and G. Pappas, "Multi-vehicle path planning in dynamically changing environments," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, may 2009, pp. 2449–2454. [Online]. Available: <http://ieeexplore.ieee.org/document/7076321/?arnumber=7076321http://ieeexplore.ieee.org/document/5152520/>
- [13] R. W. Beard and T. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *42nd IEEE International Conference on Decision and Control*, vol. 1. IEEE, 2003, pp. 25–30. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=1272530
- [14] J. S. Bellingham, M. Tillerson, M. Alighanbary, and J. P. How, "Cooperative path planning for multiple UAVs in dynamic and uncertain environments," in *Proceedings of the 41st IEEE Conference on Decision and Control 2002*, vol. 3, no. December. IEEE, 2002, pp. 2816–2822. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1184270>
- [15] Feng-Li Lian and R. M. Murray, "Real-time trajectory generation for the cooperative path planning of multi-vehicle systems," *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 4, no. December, pp. 3766–3769, 2002. [Online]. Available: <http://ieeexplore.ieee.org/document/1184950/http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1184950>
- [16] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-Time Motion Planning for Agile Autonomous Vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, jan 2002. [Online]. Available: <http://arc.aiaa.org/doi/10.2514/2.4856>
- [17] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.
- [18] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [19] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 1451–1456.
- [20] E. M. Vaisbord and V. I. Zhukovskii, *Introduction to Multi-player Differential Games and Their Applications*. Routledge, 1988.
- [21] J. F. Fisac and S. S. Sastry, "The pursuit-evasion-defense differential game in dynamic constrained environments," in *IEEE Conference on Decision and Control*, 2015.
- [22] M. Su, Y. Ji Wang, and L. Liu, "Bounded guidance law based on differential game for three-player conflict," in *IEEE Conference on Modeling, Identification, and Control*, 2014.
- [23] S. Tanimoto, "On a class of three-player differential games," *Journal of Optimization Theory and Applications*, vol. 25, no. 3, p. 469–473, 1978.
- [24] M. Chen, J. Fisac, C. J. Tomlin, and S. Sastry, "Safe sequential path planning of multi-vehicle systems via double-obstacle hamilton-jacobi-isaacs variational inequality," in *European Control Conference*, 2015.
- [25] M. Chen, Q. Hu, C. Mackin, J. Fisac, and C. J. Tomlin, "Safe platooning of unmanned aerial vehicles via reachability," in *IEEE Conference on Decision and Control*, 2015.
- [26] M. Chen, Z. Zhou, and C. J. Tomlin, "Multiplayer reach-avoid games via low dimensional solutions and maximum matching," in *Proceedings of the American Control Conference*, 2014.
- [27] —, "Multiplayer Reach-Avoid Games via Pairwise Outcomes," *IEEE Transactions on Automatic Control*, pp. 1–1, 2016. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7486004>
- [28] W. Lin, "Differential games for multi-agent systems under distributed information," Ph.D. dissertation, University of Central Florida, 2013.
- [29] M. Chen, J. C. Shih, and C. J. Tomlin, "Multi-vehicle collision avoidance via hamilton-jacobi reachability and mixed integer programming," in *IEEE Proc. Conf. on Decision and Control*, Dec., pp. 1695–1700.
- [30] M. G. Crandall, L. C. Evans, and P.-L. Lions, "Some properties of viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, vol. 282, no. 2, pp. 487–502, apr 1984.
- [31] O. Bokanowski, N. Forcadell, and H. Zidani, "Reachability and minimal times for state constrained nonlinear problems without any controllability assumption," *SIAM Journal on Control and Optimization*, pp. 1–24, 2010.
- [32] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Shankar, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *18th International Conference on Hybrid Systems: Computation and Controls*, 2015.
- [33] M. Chen, Q. Hu, J. Fisac, K. Akametalu, C. Mackin, and C. Tomlin, "Guaranteeing safety and liveness of unmanned aerial vehicle platoons on air highways," *arXiv:1602.08150*, 2016.